

Introduction to programming in R

Data Structures

- A format for storing data
- Different **types** and **dimensions**
- Most common data structure is the *vector*
- We will also use *matrices* and *data frames*
 - a *data frame* can be a matrix, but where the variables (columns) have names

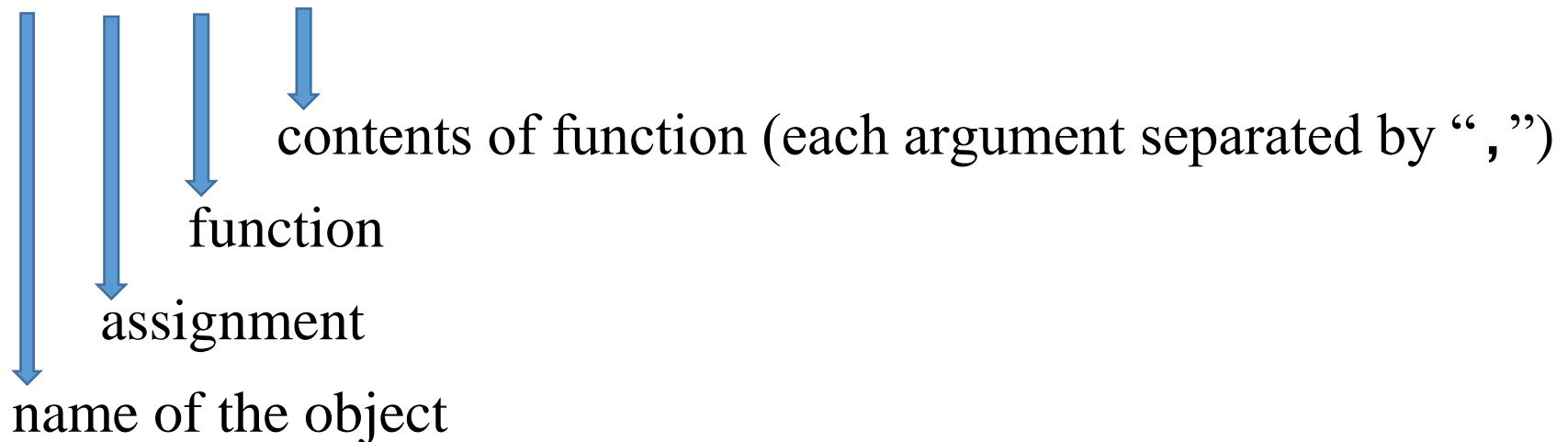
Data **types** (4 common types)

- **numeric**, or “double” – continuous, most variables in economics
- **logical** – TRUE/FALSE or 0/1 (e.g. dummy variables)
- **integers** – e.g. count data
- **character/string** – usually only arise before data is “cleaned”

Unlike a *matrix*, each variable (column) in a *data frame* can be of a different *type*.

Vectors can be created with the `c()` function (combine):

```
y <- c(2, 3.5, 8, 4)
```



We could use `=` instead of `<-` for assignment, but this is bad form.

Create this vector in R.

Functions

- There are many functions programmed in R
- They work similarly to how they work in mathematics
- In a typical function, you provide it with inputs (or arguments), R calculates something, and then provides you with output
- You can (and will) program your own functions

Let's program a function that calculates the mean of a variable, using other "base" functions.

Make sure you have first created a variable in R (in order to test our functions):

```
y <- c(2, 3.5, 8, 4)
```

Calculate the mean using the base function `mean()`:

name of function

argument/input

```
> mean(y)  
[1] 4.375
```

output

To create our own function to calculate the mean, we need other base functions in R.

- What three functions do we need?

- Summation: `sum()`
- Division: `"/"()`
- `length()`

Note:

`"/"(4, 2)`

Is equivalent to:

`4 / 2`

(R makes it easier to “call” some functions, in an intuitive way)

Test the function to make sure it does the same thing as R's `mean()`:

```
> mymean(y)
[1] 4.375
```

Use it for any arbitrary vector:

```
mymean(c(1, 2, 3, 4))
```

Assignment – worth 2% of project - due Tues. Sept. 18th

Write a function that calculates the sample variance of a variable, without using the `var()` function in R.

Hints:

- to “square” every element in a vector use “`^ 2`”
- computers do mathematical operations in a particular order. To control the order, use parentheses. For example:

`3 + 2 ^ 2`

gives a different result from

`(3 + 2) ^ 2`

Turn in a single printed out sheet containing your R code, which includes your function, and how you tested it to get the same result as `var()`.

Do not use `y <- c(2, 3.5, 8, 4)` to test your code. Use anything else.

For example, the above would be submitted in assignment form as:

```
y <- c(2, 3.5, 8, 4)
mean(y)
```

```
mymean <- function(x) {
  sum(x) / length(x)
}
mymean(y)
```

With a copy of the R output obtained from running the above code:

```
> y <- c(2, 3.5, 8, 4)
> mean(y)
[1] 4.375
>
> mymean <- function(x) {
+   sum(x) / length(x)
+ }
> mymean(y)
[1] 4.375
```